REMARKS

In regard to the Examiner's claim rejections under 35 USC Article 103, the Examiner has rejected claims 1 to 40 for obviousness under 35 USC 103(a) as being unpatentable over *Phanorious et al.* (Transforming Command-Line Driven Systems to Web Applications) in view of *Eagar et al.* (U.S. Patent 5,969,200).

At this juncture, Applicant would traverse the Examiner's considerations that the above rejected claims are obvious in view of *Phanorious* when combined with *Eagar*.

As submitted to the Examiner in previous responses, the claimed invention provides a method and system which receives legacy source code relating to a user interface with the functionality to interact with a legacy application (e.g. a user interface) and utilizes the legacy code to produce a series of executable software components which provide necessary functionality for interacting with the legacy software application.

A minor clarifying amendment has been made to each of the independent claims, to more clearly describe the claimed invention. In particular, the independent claims now recite the feature of <u>providing a software application which</u> utilizes the legacy source code to <u>automatically</u> produce a series of executable software components.

Phanorious teaches a software application that allows a user or a programmer to <u>manually create</u> a graphical user interface for a legacy application that utilizes a command driven interface.

The Examiner's attention is drawn to page 6 of the *Phanorious* disclosure, which clearly states that the graphical user interface (GUI) of *Phanorious* is generated "from an *application description* and a set of *UI mapping rules*, <u>generated by a person familiar with how to run the application</u>" (underlining added).

As submitted to the Examiner in a previous response, the presently claimed invention <u>generates</u> the required executable software components when provided with legacy code.

There is no disclosure in *Phanorious* of automatically generating executable software components when provided with legacy code. Rather, *Phanorious* merely provides a user with a software tool which allows a user or developer to write UI mapping rules and an application description which may then be compiled to generate a user interface which, at the "back end", provides the legacy application with user commands in the traditional manner.

Now, as to the *Eager* reference, there is disclosed a method for transitioning a business enterprise from a single platform to a distributed platform (i.e. multiple computing systems). The method generally comprises the steps of organizing and prioritizing the transitioning of each application in the business enterprise, and rewriting or substituting certain applications used by the business enterprise.

Eagar <u>does not teach</u> a system which receives legacy source code relating to a legacy application and utilizing the legacy code to produce a series of executable software components which provide necessary functionality for interacting with the

legacy software application. Indeed, *Eagar* <u>teaches</u> <u>away</u> from such a notion, since it is concerned with transitioning an <u>entire</u> enterprise (*i.e.* both the "back end" legacy application and the "front end" user interface).

While Examiner refers to structures 118, col.23, lines 35-45 and col. 9, lines 45-65, the text at these reference points merely describe a system which takes a user interface structure (as created by a programmer) and utilizes the structure to produce a graphical user interface. There is no description of utilizing <u>legacy source code</u> to produce a series of executable software components.

Moreover, it is not clear how this reference reads onto *Phanorious*, nor is there any suggestion or teaching in *Phanorious* that *Eager* should or would be combined to arrive at the claimed invention.

As was previously stated, it should be indicated that it is improper for the Examiner to combine a second reference with a first reference if there is no indication in the first reference that such technology from the second reference would be desirable.

For example, in the case of *In re Jones*, 958 Fed.2d, p.347, 21 USPQ2d, pp.1941,1943 (Fed.Cir 1992), it was stated as follows:

> Before the PTO may combine the disclosures of two or more prior art references in order to establish *prima facie* obviousness, there must be some suggestion for doing so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art.

Moreover, in the case of *Uniroyal, Inc. v. Rudkin-Wiley Corp.*, 837 F.2d p.1044, and 5 USPQ2d, p.1434 (Fed.Cir 1988), it was stated:

> When prior art references require a selected combination to render obvious a subsequent invention, there must be some reason for the combination other than the hindsight gleaned from the invention itself. Something in the prior art as a whole must suggest desirability, and thus, the obviousness of making the combination. It is impermissible to use the claims as a frame and the prior art references as a mosaic to piece together a facsimile of the claimed invention.

The Examiner has not indicated or pointed out which statements or paragraphs in *Phanorious* teach the desirability of generating a user interface utilizing legacy source code, or alternatively, which statement teaches or suggests that it would be desirable to combine *Eagar* with *Phanorious*. Applicant respectfully submits that there is <u>no suggestion</u> in *Phanorious* that it would be desirable to generate, a user interface utilizing legacy source code. Indeed, *Phanorious*, at pages 7 and 8 generally clearly states that the entire interface (*i.e.* the layout of controls on the interface, and the functions that each of the controls perform) are to be <u>generated by a user (programmer) who has knowledge of the input commands of the legacy software application</u>.

In view of the situation that the cited references to *Phanorious* and *Eagar* cannot be considered having the teachings which would cover Applicant's independent claims, and further in regard to the impropriety of combining such said references to try to re-invent Applicant's method and system from enumerable cited references, it should be understood that Applicant's

claims should be taken as a whole in their entirety, where as such, they indicate a functionality which cannot be provided for by the cited references, neither in the independent claims, nor in the dependent claims involved herein.

As a result, it is respectfully requested that Examiner consider Applicant's claims as a whole in their entirety and subsequently provide a timely Notice of Allowance.

Respectfully submitted,

By: _Alfred W. Kozak_
Alfred W. Kozak
Reg No. 24,265
(858) 451-4615
(949) 380-5822

IN THE DRAWINGS:

        Formal drawings have been previously sent.